logical and physical portion. It depicts the use of a common default/prototype interface and the use of a complementary physical level scheme. Together, the logical scheme and the physical scheme form a complete description of a given instance of the component prototype.

[0017] For the sake of simplicity, this figure omits the ability of any given component to be composed of an arbitrary assembly of subordinate components.

Example—Power Supply

[0018] Various aspects of the present invention are described in the context of a particular example—a power supply. Those skilled in the art will realize that this example is not meant to limit the scope of the invention in any way, and is given for explanatory purposes.

[0019] A power supply is a component of every computer system and of every peripheral device. The Power Supply example used here illustrates how component assemblies can be affected via one particular default/prototype scheme used for a wide range of (electrical) "PowerSupply" components. Power supplies come in many different flavors. Some have self-contained intelligent management sub-systems, while others do not. Some power supplies are actually power supply systems with multiple power supply units within them. The range of specific power supply types is surprisingly long. According to embodiments of the present invention, however, all these specific power supply types may be treated as some sort of generic, logical (electrical) power supply. Consider, for example, the electrical power supply of a peripheral device.

[0020] Generic methods are parameterized with both logical and physical power supply descriptions. The prototype has flexible implementations for each behavioral facet/method/slot. As used herein, a "facet" is defined to be either a reference to some operational (run-time) behavior or a reference to a certain attribute/property/datum (about the current state of the corresponding peripheral hardware device). It is preferably possible for any behavior/method/slot to be substituted for any other. According to embodiments of the present invention, it is possible for one logical instance to be composed of a graph of subordinate logical instances (according to the scheme for peripheral in question).

[0021] Interfaces for any given general type are required to carry a certain expected number of software/firmware interface facets. As noted above, peripheral hardware devices run a gamut from fans to sensors (of various increasingly specialized types), and include LEDs, LCDs (Liquid-Crystal Displays), modems, NICs (Network Interface Cards) (or individual MAC/PHY chips) and more.

[0022] Returning to the example of a certain power supply device, it is expected that all electrical power supplies will support the following logical facets:

[0023] 1. Power Initialization (possibly a "do-nothing" facet if nothing is required);

[0024] 2. Power Information (power related information in a certain data format);

[0025] 3. Power On (activate/start/turn-on);

[0026] 4. Power Off (de-activate/stop/turn-off);

[0027] 5. Power Reset;

[0028] 6. Power Cycle (typically (a) power-on, (b) pause, (c) power-off); and

[0029] 7. Power Status (the current power status).

[0030] A corresponding physical description/implementation to drive the implementation logic behind these logical facets is provided below, along with a description of how these logical facet implementations eventually bind to physical electrical signaling interfaces. For the purposes of this illustration, focus on the following select list of expected power supply (sub-system) facets.

[0031] For the sake simplicity of this illustration, further/additional interface facets (i.e., features/functions) that certain, particular and/or specific types of power supplies might (or might not) provide are not described. Besides, this restraint can contribute to the range of power supply peripherals can be modeled with this particular scheme. Importantly, the present invention circumvents the need for any more specific types of power supply models.

[0032] The present invention generally requires that any associated peripheral (and/or host) device be described (both logically and physically) within an embedded software/firmware part. When there is an associated power supply (peripheral device), then the embedded software/firmware part must populate a certain logical power supply description. The following is an example of a particular logical schema used to describe all (electrical) power supplies:

[0033] Logical Power On Signal (to turn power on)

[0034] Logical Power On Signal Width (i.e., duration)

[0035] Logical Power Off Signal (to turn power off)

[0036] Logical Power Off Signal Width (i.e., duration)

[0037] Logical Power Reset Signal (to reset power)

[0038] Logical Power Reset Signal Width (i.e., duration)

[0039] Logical Power Status (to read current power status—true/false state)

[0040] Logical Interrupt Request (IRQ) type (for external button push event)

[0041] Logical Interrupt Request number (an IRQ discriminator)

[0042] Logical Interrupt Request Interpretation (activation, de-activation or both)

[0043] <Logical Descriptors of Behaviors> (e.g., function/method signatures)

[0044] Each of the enumerated descriptive attributes parameterizes the <Logical Descriptors of Behaviors>. Each of these enumerated descriptive attributes appears (packed into) a certain block of bits ("BoB"—described in greater detail below). One implementation of the <Logical Descriptors of Behaviors> is illustrated in **FIG. 2**.

[0045] As used herein, the structural conventions of each logical description are referred to as a "scheme" (singular). The terms "schema" (plural) and "schemata" (plural) are used to refer to some collection of schemataes. (So the present example describes a Power Supply scheme.) Com-